

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



LẬP TRÌNH NÂNG CAO

BÀI TẬP LỚN
WEB TRA CỨU SAO KÊ

Lớp: L01

Nhóm: 492

GVHD: Lê Đình Thuận

Danh sách thành viên: Nguyễn Quang Huy- 2311202

Nguyễn Quang Tùng - 2313817

Nguyễn Phú Vinh - 2313922

Lương Ngọc Trung - 2313668

Huỳnh Xuân Quốc Việt - 2313891

Trịnh Trần Trung Tín - 2313462

TP. HỒ CHÍ MINH, THÁNG 12/2024

MỤC LỤC

I. GIỚI THIỆU	3
II. PHÂN TÍCH VÀ THIẾT KẾ	4
1. Phân tích yêu cầu	4
2. Sơ đồ luồng dữ liệu	4
a) Input (Đầu vào):	4
b) Xử lý (Processing):	4
c) Output (Đầu ra):	4
Hình 2.1: Sơ đồ luồng dữ liệu (DFD).....	5
3. Giao diện hiển thị	5
Hình 2.2: Giao diện trang chủ ở Day mode	5
Hình 2.3: Giao diện trang chủ ở Night mode	6
Hình 2.4: Giao diện khi tìm kiếm.....	6
Hình 2.5: Giao diện ở cuối trang tìm kiếm.....	7
Hình 2.6: Giao diện lịch sử tìm kiếm	7
Hình 2.7: Giao diện khi chưa có lịch sử tìm kiếm	8
III. QUY TRÌNH TRIỂN KHAI	8
1. Xây dựng Front-End.....	8
1.1. Thiết kế giao diện người dùng.....	8
1.2. Tối ưu hoá responsive	9
2. Phát triển Back-End	9
2.1. Kiến trúc hệ thống.....	9
2.2. Công nghệ sử dụng	9
2.3. Tối ưu hiệu năng.....	11
IV. CÁCH CHẠY WEBSITE.....	12
V. SOURCE CODE	12

I. GIỚI THIỆU

Vừa qua, do tình hình bão số 3 Yagi đã trở nên nghiêm trọng ảnh hưởng tới đời sống của người dân miền Bắc trầm trọng, chính vì thế người dân cả nước đã đồng lòng hướng đến miền Bắc để đùm bọc chia sẻ lẫn nhau. Người có điều kiện, thời gian thì cùng nhau mua gạo, nhu yếu phẩm và chất lên xe tiến đến vùng bão lũ. Vậy những người không có thời gian thì sao? Mặt trận Tổ Quốc Việt Nam đã vô cùng hiểu chuyện nên đã giúp đỡ người dân bằng cách cho người dân chuyển tiền vào sổ tài khoản của Mặt trận Tổ Quốc Việt Nam và số tiền ấy sẽ đến tận tay người dân vùng bão lũ khó khăn. Người có điều kiện thì chuyển nhiều, người không có điều kiện thì chuyển ít. Mỗi người một ít tạo thành một khối đồng nhất hùng mạnh cùng nhau vượt qua cơn bão, thể hiện tinh thần đoàn kết và sức mạnh của dân tộc Việt Nam. Tuy nhiên, có những người đã lợi dụng điều này để “căng phong kẻ bắt”, đẩy tương tác của mình trên MXH. Để minh bạch, Mặt trận Tổ Quốc Việt Nam đã đi một nước đi khiến những người “phông bạt” ấy không thể nào ngờ. MTTQVN đã công bố danh sách toàn bộ số tiền được nhận thông qua chuyển khoản qua STK của MTTQ. Cư dân mạng đã vô cùng tinh mắt và nhạy bén đã dò ra những người đã lợi dụng điều này mà làm lợi cho bản thân. Chính vì thế trang web của chúng tôi đã được ra đời để có thể hỗ trợ mọi người trong việc tìm kiếm và tra cứu sao kê của MTTQVN dễ dàng và thuận tiện. Đây là một công cụ trực tuyến, đơn giản nhưng hiệu quả, giúp người dùng nhanh chóng tìm kiếm thông tin chi tiết về các khoản đóng góp.

- **Mục tiêu:** Trang web hướng tới việc tăng cường tính minh bạch, thuận tiện và chính xác trong việc quản lý thông tin sao kê, đồng thời giúp các cá nhân, tổ chức có thể dễ dàng kiểm tra giao dịch của mình mà không cần phải truy cập trực tiếp ngân hàng.

- **Công nghệ sử dụng:** Trang web sử dụng những thành phần cơ bản để xây dựng một trang web

Frontend: Ngôn ngữ sử dụng là HTML, CSS và JavaScript

Backend: Ngôn ngữ sử dụng là Node.js

Module sử dụng là express.js, node-cache, ...

Công cụ phát triển: Visual Studio Code, Github.

II. PHÂN TÍCH VÀ THIẾT KẾ

1. Phân tích yêu cầu

Các chức năng chính:

- Tra cứu sao kê theo ngày, số tiền hoặc nội dung giao dịch.
- Hiện thị kết quả chi tiết theo và cho phép sắp xếp nội dung.
- Cho phép xem lại lịch sử những gì đã tìm kiếm.
- Thông báo khi không có kết quả tìm kiếm.
- Xuất file PDF trang hiện tại.
- Chuyển sang chế độ tối hoặc sáng

2. Sơ đồ luồng dữ liệu

a) Input (Đầu vào):

- Người dùng nhập các trường tìm kiếm (ngày, số tiền, nội dung) qua giao diện web
- Frontend lưu trữ lịch sử tìm kiếm trong Local Storage
- Request được gửi qua HTTP GET đến Express server

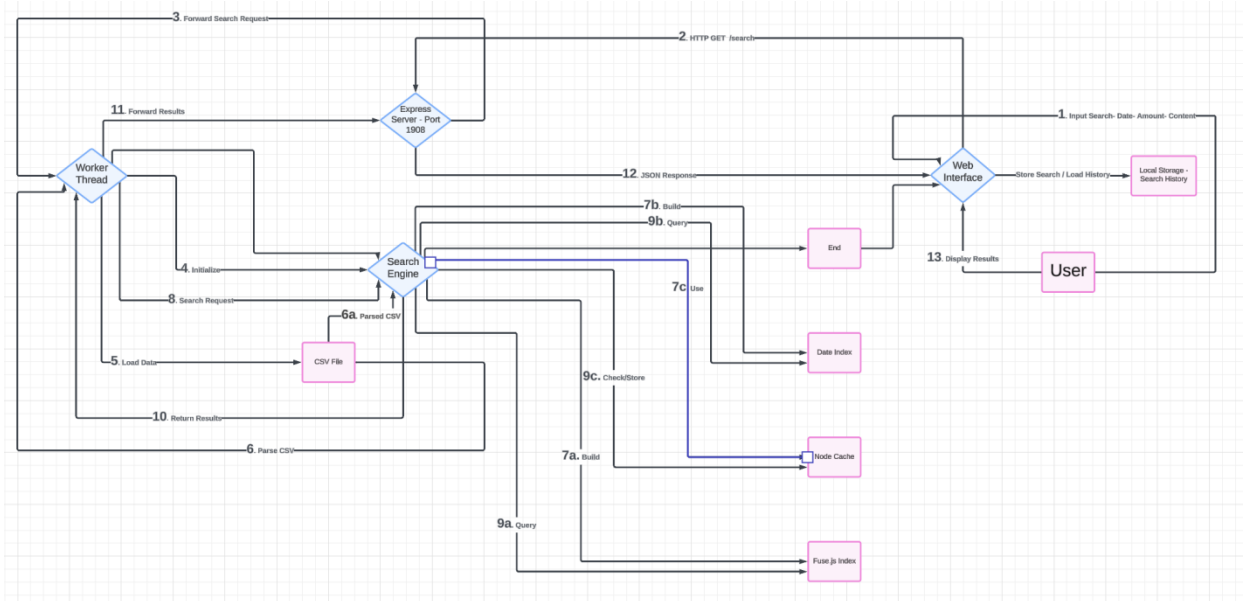
b) Xử lý (Processing):

- Server nhận request và chuyển tiếp đến Worker Threads
- Worker Threads:
 - + Khởi tạo SearchEngine
 - + Đọc và parse file CSV sử dụng Paparse
 - + Chuyển dữ liệu đã parse cho SearchEngine
- SearchEngine:
 - + Xây dựng các index:
 - Date Index: index theo ngày để tìm kiếm nhanh
 - Fuse.js Index: index full-text search cho nội dung
 - + Sử dụng NodeCache để cache kết quả tìm kiếm
 - + Thực hiện tìm kiếm dựa trên các tiêu chí đã nhập
 - + Sắp xếp và phân trang kết quả tìm được

c) Output (Đầu ra):

- SearchEngine trả kết quả về Worker Threads
- Worker Threads chuyển kết quả về Express Server
- Server trả về JSON response cho Frontend
- Frontend hiện thị kết quả cho người dùng và cập nhật phân trang.

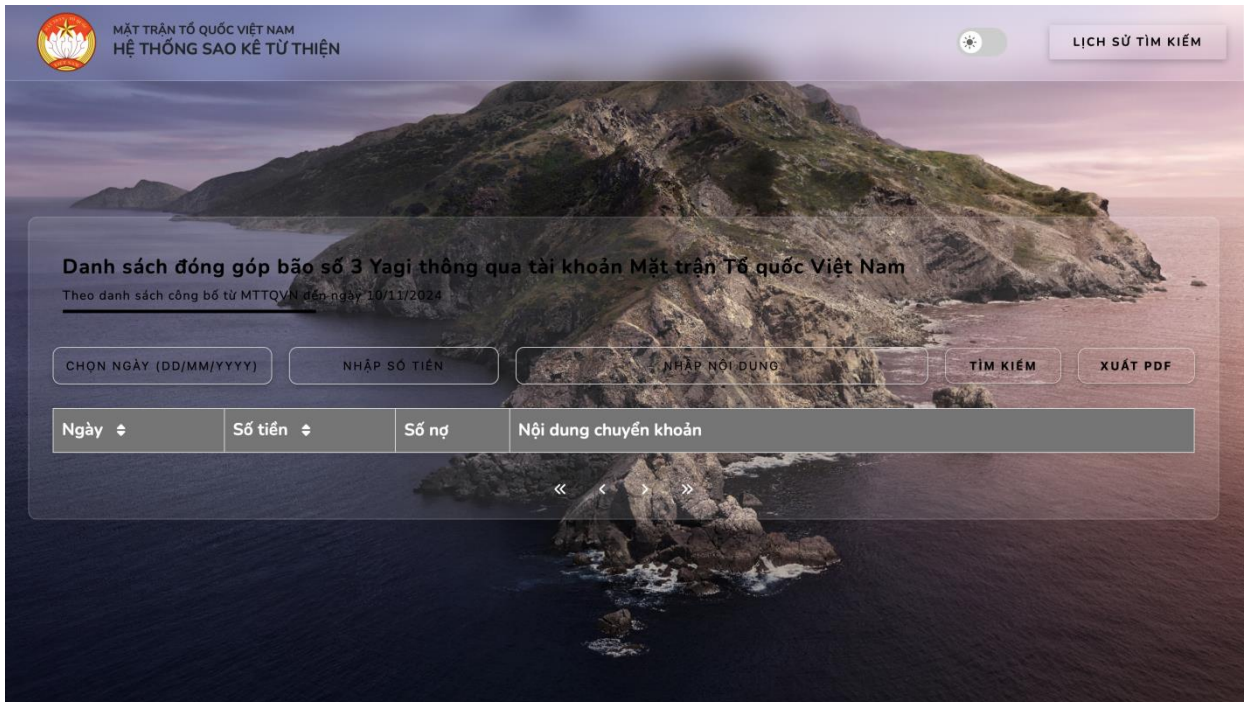
- Dưới đây là sơ đồ luồng dữ liệu (DFD) của trang web:



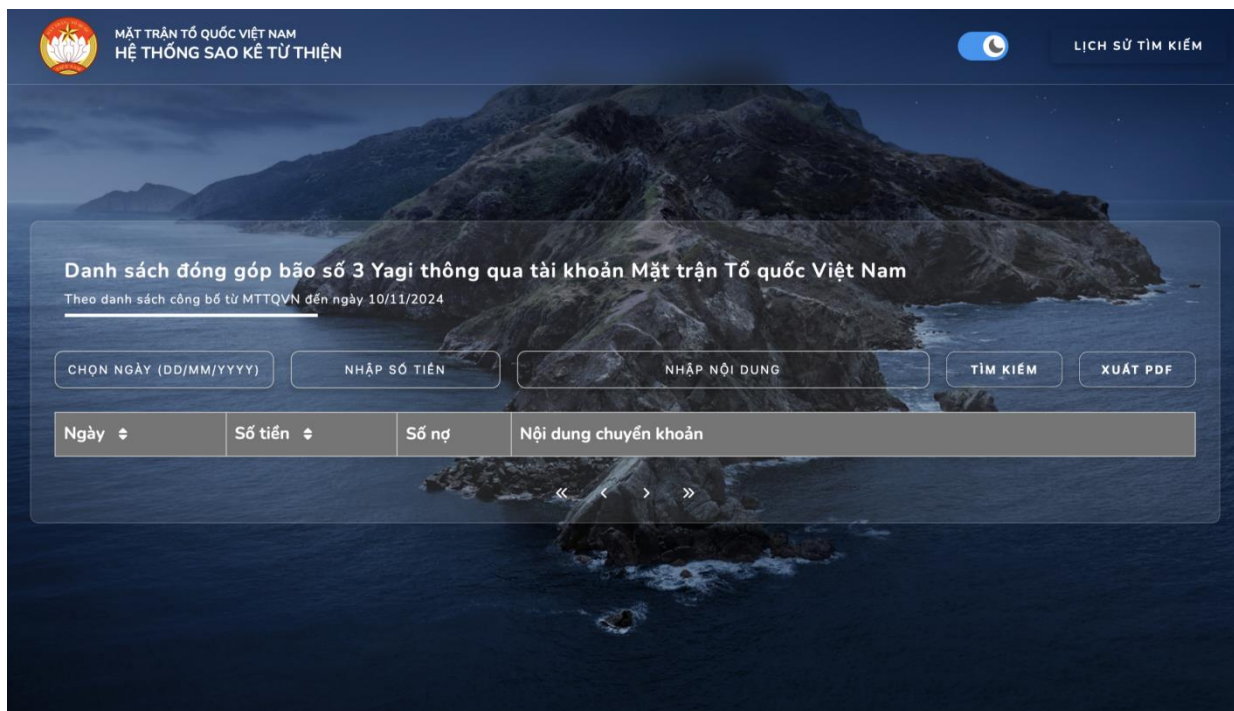
Hình 2.1: Sơ đồ luồng dữ liệu (DFD)

3. Giao diện hiển thị

- Đây là giao diện hiển thị đầu tiên khi truy cập vào trang web, bao gồm tính năng tìm kiếm, xem lại lịch sử, xuất file pdf và chuyển chế độ hiển thị:

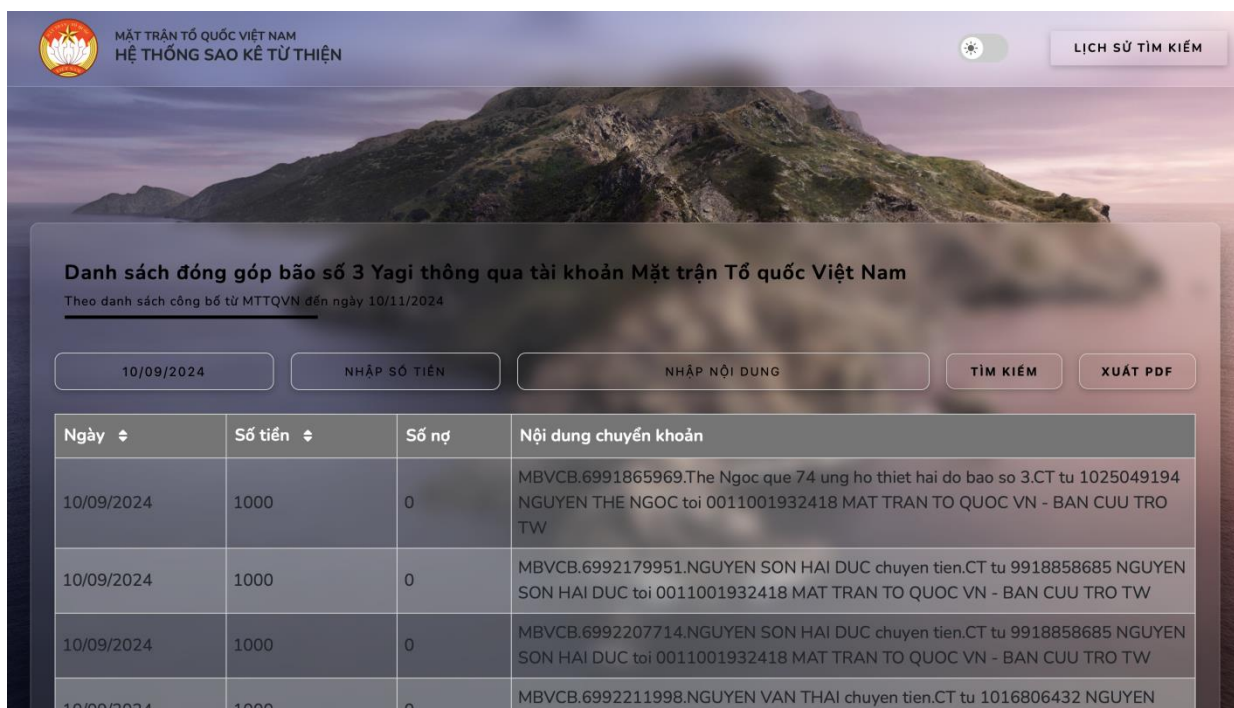


Hình 2.2: Giao diện trang chủ ở Day mode



Hình 2.3: Giao diện trang chủ ở Night mode

- Khi tìm kiếm theo các trường (ngày, số tiền, nội dung), bảng kết quả sẽ hiện lên cho phép xem chi tiết các thông tin tìm kiếm, sắp xếp theo các trường theo ngày hoặc số tiền và chuyển trang:




Hình 2.4: Giao diện khi tìm kiếm

QUOC VN - BAN CUU TRO TW			
10/09/2024	1000	0	MBVCB.6996723620.NGUYEN DANG HOA . UNG HO KHAC PHUC HAU QUA BAO SO 3 YAGI.CT tu 1036589691 NGUYEN DANG HOA toi 0011001932418 MAT TRAN TO QUOC VN - BAN CUU TRO TW
10/09/2024	1000	0	MBVCB.6996750048.Tro ly ca si Ha Nhi ung ho dong bao mien Bac ruot thit.CT tu 0191000361411 NGO HA BAO VY toi 0011001932418 MAT TRAN TO QUOC VN - BAN CUU TRO TW
10/09/2024	1000	0	MBVCB.6996762289.gop phan ho tro bao Yagi.CT tu 1015204666 PHAN LAI CONG DAT toi 0011001932418 MAT TRAN TO QUOC VN - BAN CUU TRO TW
10/09/2024	1000	0	MBVCB.6996779367.TRAN HOANG NHAT LAM chuyen tien.CT tu 0121000887888 TRAN HOANG NHAT LAM toi 0011001932418 MAT TRAN TO QUOC VN - BAN CUU TRO TW
10/09/2024	1000	0	MBVCB.6996886037.NGUYEN HOANG TUAN ung ho lu lut mien bac do bao cogi.CT tu 1043435764 NGUYEN HOANG TUAN toi 0011001932418 MAT TRAN TO QUOC VN - BAN CUU TRO TW
10/09/2024	1000	0	MBVCB.6997127088.LE HUYNH QUANG DONG chuyen tien.CT tu 1028959148 LE HUYNH QUANG DONG toi 0011001932418 MAT TRAN TO QUOC VN - BAN CUU TRO TW
10/09/2024	1000	0	MBVCB.6997156356.LE HUYNH QUANG DONG chuyen tien.CT tu 1028959148 LE HUYNH QUANG DONG toi 0011001932418 MAT TRAN TO QUOC VN - BAN CUU TRO TW

Hình 2.5: Giao diện ở cuối trang tìm kiếm

- Trang web còn cho phép người dùng xem lại lịch sử những gì đã tìm kiếm:



MẬT TRẦN TỔ QUỐC VIỆT NAM
HỆ THỐNG SAO KÊ TỬ THIÊN

LỊCH SỬ TÌM KIẾM

Danh sách đóng góp bảo số 3 Yagi thông qua tài khoản Mật trần Tổ quốc Việt Nam

Theo danh sách công bố từ MTTQ

10/09/2024

Lịch sử tìm kiếm

Ngày: 10/09/2024

12:25:32 5/12/2024

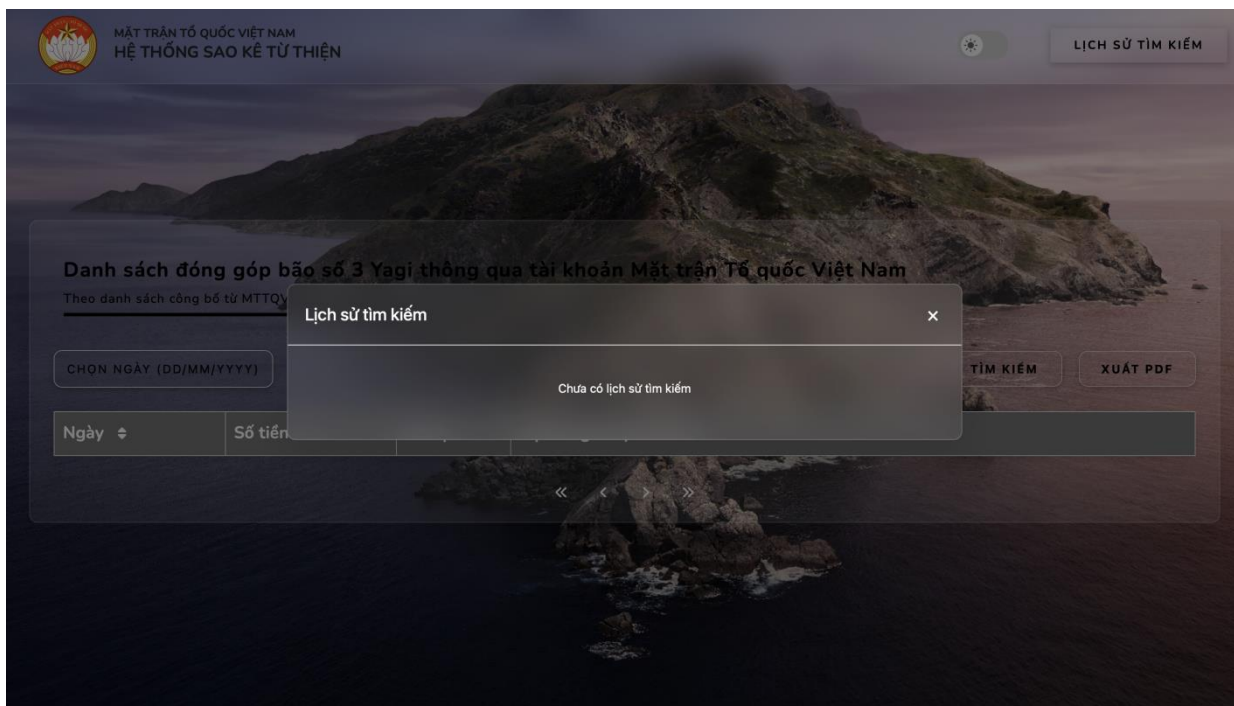
TÌM KIẾM

XUẤT PDF

Ngày	Số tiền		
10/09/2024	1000	0	MBVCB.6991865969.The Ngọc que 74 ung ho thiet hai do bao so 3.CT tu 1025049194 NGUYEN THE NGOC toi 0011001932418 MAT TRAN TO QUOC VN - BAN CUU TRO TW
10/09/2024	1000	0	MBVCB.6992179951.NGUYEN SON HAI DUC chuyen tien.CT tu 9918858685 NGUYEN SON HAI DUC toi 0011001932418 MAT TRAN TO QUOC VN - BAN CUU TRO TW
10/09/2024	1000	0	MBVCB.6992207714.NGUYEN SON HAI DUC chuyen tien.CT tu 9918858685 NGUYEN SON HAI DUC toi 0011001932418 MAT TRAN TO QUOC VN - BAN CUU TRO TW
10/09/2024	1000	0	MBVCB.6992211998.NGUYEN VAN THAI chuyen tien.CT tu 1016806432 NGUYEN

Hình 2.6: Giao diện lịch sử tìm kiếm

- Khi không có nội dung hay số tiền đã nhập, giao diện sẽ hiển thị không có kết quả, tương đương với khi chưa tìm kiếm gì thì lịch sử tìm kiếm sẽ không hiện gì:



Hình 2.7: Giao diện khi chưa có lịch sử tìm kiếm

III. QUY TRÌNH TRIỂN KHAI

1. Xây dựng Front-End

1.1. Thiết kế giao diện người dùng

Bố cục tổng thể:

- Header: Logo Mặt trận Tổ quốc Việt Nam, tiêu đề hệ thống, nút chuyển chế độ và lịch sử tìm kiếm
- Phần tìm kiếm: Tích hợp bộ lọc theo ngày, số tiền và nội dung
- Bảng hiển thị kết quả: Hiển thị thông tin giao dịch theo các cột ngày, số tiền, số nợ và nội dung
- Footer: Phân trang và điều hướng kết quả

Màu sắc và phong cách:

- Sử dụng font chữ Nunito để đảm bảo tính chuyên nghiệp và dễ đọc
- Thiết kế tối giản, tập trung vào chức năng chính

Hiệu ứng tương tác:

- Animation cho date picker khi click vào ô ngày
- Highlight kết quả tìm kiếm khớp với từ khóa
- Hiệu ứng hover trên các nút bấm và dòng kết quả
- Modal hiển thị lịch sử tìm kiếm với animation

1.2. Tối ưu hoá responsive

- Thiết kế CSS Flexbox và Grid để layout tự động điều chỉnh
- Các thành phần UI tự động co giãn theo kích thước màn hình
- Bảng kết quả có khả năng cuộn ngang trên thiết bị nhỏ
- Các breakpoint chính:
 - + Desktop: >1024px;
 - + Tablet: 768px - 1024px;
 - + Mobile: <768px;

2. Phát triển Back-End

2.1. Kiến trúc hệ thống

Mô hình Worker Threads:

- Main thread: xử lý http request và response
- Worker thread: xử lý tìm kiếm và phân tích dữ liệu
- Giao tiếp giữa các thread thông qua message passing

Xử lý dữ liệu:

- Đọc file csv sử dụng thư viện Papaparse
- Phân tích và chuẩn hoá dữ liệu tự động
- Xây dựng các chỉ mục cho tìm kiếm

2.2. Công nghệ sử dụng

Node.js và Express:

- Routing và xử lý HTTP request
- Static file serving
- Body parsing và URL encoding
- Error handling

Fuse.js cho tìm kiếm mờ (fuzzy search):

- Fuse.js là thư viện hỗ trợ tìm kiếm mờ, cho phép tìm kiếm linh hoạt ngay cả khi người dùng nhập từ khoá không hoàn toàn chính xác.

- Cấu hình tìm kiếm :

```

this.searchIndex = new Fuse(data, {
  keys: [
    { name: "detail", weight: 2 },
    { name: "credit", weight: 1 },
    { name: "debit", weight: 1 },
  ],
  threshold: 0.1,
  distance: 50,
  ignoreLocation: true,
  useExtendedSearch: true,
});

```

Hình 3.1: Cấu hình tìm kiếm Fuse.js

- Keys bao gồm các trường trong dữ liệu data mà Fuse.js tìm kiếm (detail, credit, debit)
- Weight : trọng số ưu tiên của trường, trường có trọng số cao hơn sẽ được ưu tiên hơn trong quá trình tìm kiếm
- Threshold : Giá trị 0.1 có nghĩa là Fuse.js sẽ chỉ trả kết quả có mức độ tương đồng cao (90%) . Các giá trị càng gần 1.0 thì tìm kiếm sẽ càng “thoải mái” với lỗi sai.
- Distance : Giá trị 50 chỉ định khoảng cách tối đa (theo số ký tự) giữa các từ khoá trong văn bản mà Fuse.js cho phép để coi là tương đồng.
- Ignore Location :
 - + True nghĩa là vị trí của từ khoá trong chuỗi không quan trọng (ví dụ từ khoá ở đầu hay cuối chuỗi vẫn được tính)
 - + False sẽ yêu cầu từ khoá xuất hiện gần nhau hơn
- useExtendedSearch : cho phép tìm kiếm nâng cao (kích hoạt các toán tử tìm kiếm đặc biệt của Fuse.js, hỗ trợ tìm kiếm không phân biệt hoa thường)

Node – cache cho bộ nhớ đệm

- Node – cache được sử dụng để lưu trữ dữ liệu tạm thời giúp tăng tốc độ xử lý (vì không cần truy xuất dữ liệu từ nguồn khác), giảm tải cho hệ thống và tự động xóa dữ liệu không còn cần thiết sau một khoảng thời gian.

- Cấu hình cache :

```

this.cache = new NodeCache({
  stdTTL: 1200,
  checkperiod: 600,
  maxKeys: 10000,
  useClones: false,
});

```

Hình 3.2: Cấu hình Node - cache

- Standard Time to Live (stdTTL): Thời gian tồn tại mặc định của dữ liệu trong cache. Giá trị ở đây là 1200 (20 phút).
- Checkperiod: Chu kỳ kiểm tra. Node – cache sẽ và xoá dữ liệu hết hạn mỗi 10 phút giúp giải phóng bộ nhớ và không gây quá tải cho hệ thống
- maxKeys: Số lượng khoá tối đa mà cache có thể chứa
- useClones:
 - + False: Cache sẽ lưu trữ tham chiếu trực tiếp đến dữ liệu ban đầu
 - + True: NodeCache sẽ tạo bản sao của dữ liệu trước khi lưu, giúp tránh thay đổi dữ liệu gốc bên ngoài cache.
- Lợi ích của sử dụng Node – cache và các cấu hình trên:

Hiệu quả bộ nhớ:

- Với thời gian tồn tại stdTTL và chu kỳ kiểm tra checkperiod, NodeCache đảm bảo không giữ lại dữ liệu quá lâu, giải phóng bộ nhớ cho các tác vụ khác.

Tối ưu hiệu suất:

- Cache hoạt động trong bộ nhớ (RAM), giúp truy cập dữ liệu nhanh hơn nhiều so với đọc từ cơ sở dữ liệu hoặc file.

Giới hạn số lượng khóa (maxKeys):

- Đảm bảo cache không vượt quá kích thước bộ nhớ, đặc biệt trong hệ thống có tài nguyên hạn chế.

Kiểm soát sao chép dữ liệu (useClones):

- Tắt chế độ sao chép dữ liệu (false) giúp tăng tốc độ lưu trữ và giảm tiêu tốn bộ nhớ, đặc biệt khi dữ liệu lớn.

2.3. Tối ưu hiệu năng

Indexing:

- Xây dựng chỉ mục theo ngày (dateIndex)
- Dữ liệu được phân nhóm và lưu trữ trong dateIndex
- Khi tìm kiếm theo ngày, chỉ cần truy cập trực tiếp vào this.dateIndex[date], thay vì duyệt toàn bộ dữ liệu.
- Xây dựng chỉ mục song song (Parallel Indexing)

- Chỉ mục ngày và chỉ mục tìm kiếm được xây dựng song song bằng cách sử dụng Promise.all() trong phương thức buildIndices().
- Hai quá trình không phụ thuộc lẫn nhau:
- Chỉ mục ngày (buildDateIndex): Chia nhóm dữ liệu theo ngày.
- Chỉ mục tìm kiếm (buildSearchIndex): Tạo công cụ tìm kiếm với Fuse.js.

Cache:

- Cache kết quả tìm kiếm phổ biến
- Tự động invalidate cache khi hết hạn
- Quản lý memory footprint.

IV. CÁCH CHẠY WEBSITE

Bước 1: Điều hướng về dự án

- cd WebSaoKe

Bước 2: Download Node.js

- Kiểm tra xem Node.js đã được cài đặt chưa: node -v
- Nếu chưa, cài đặt homebrew (nếu đã có homebrew bỏ qua) :

```
/bin/bash -c "$(curl -fsSL
```

[https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh\)"](https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh))

- Kiểm tra xem homebrew đã được cài đặt chưa: brew -v
- Cài đặt Node.js: brew install node
- Kiểm tra phiên bản node.js và npm:

```
node -v
```

```
npm -v
```

Bước 3: Download các module cần thiết

```
npm install express body-parser papaparse node-cache fuse.js nodemon
```

Bước 4: Chạy dự án

```
nodemon app.js
```

Bước 5: Truy cập đường link sau : http://localhost:1908

V. SOURCE CODE

- Link github, phần source code nằm trong nhánh main:
<https://github.com/chuntinne05/WebSaoKe.git>
- Link nhóm đã deploy:
<https://hcmutsaoke.onrender.com>